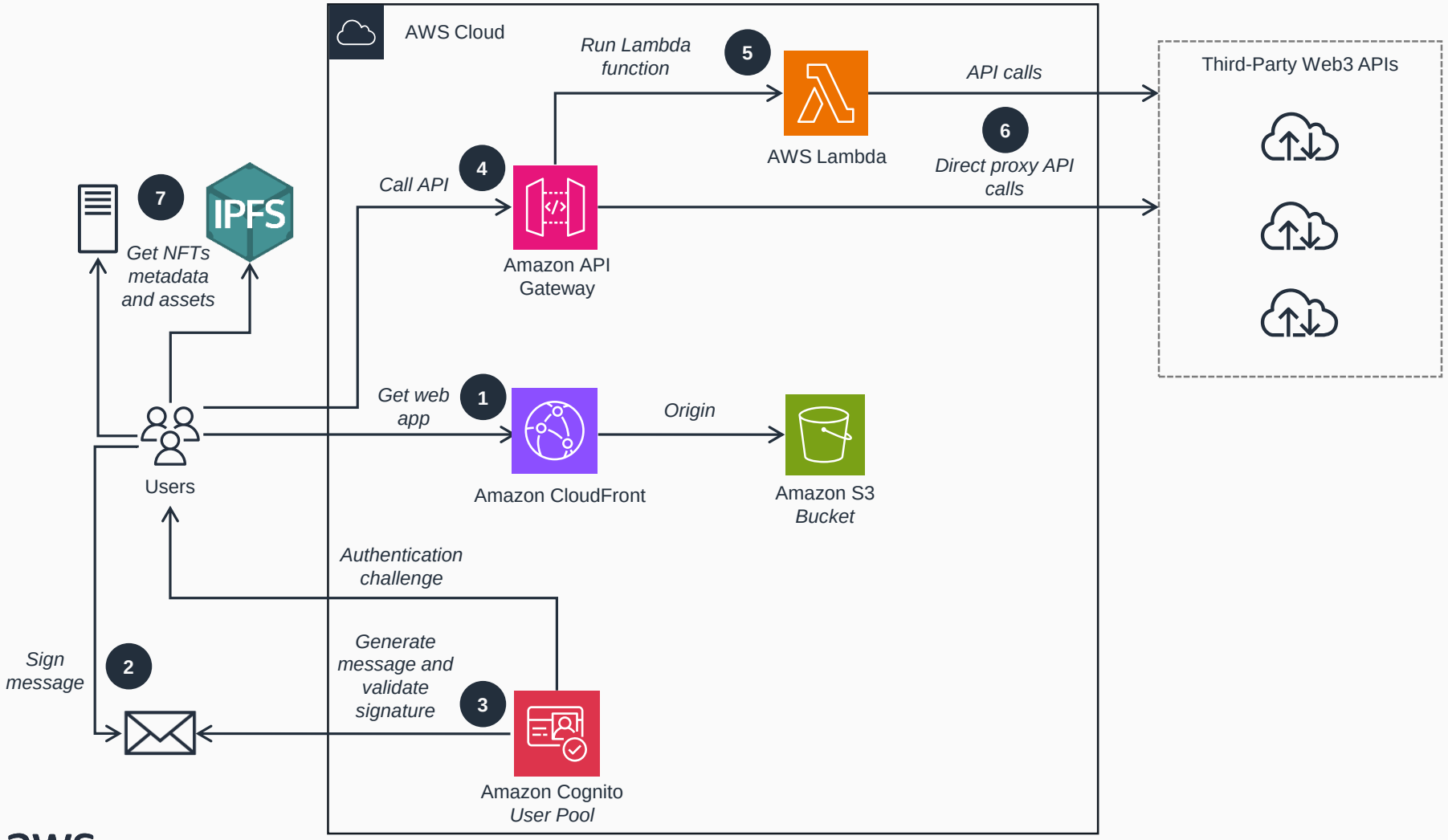
Guidance for Authentication with Digital Wallets on AWS

This architecture diagram enables you to authenticate with your digital wallet and obtain AWS credentials. Those credentials can be used to access AWS services and to make secure API calls to third-party Web3 APIs.



1. Download the dApp from the web by connecting to the **Amazon CloudFront** distribution endpoint, which uses an **Amazon Simple Storage Service (Amazon S3)** bucket as Origin.
2. You will be asked to sign a generated message using your digital wallet and private key. Then, the dApp sends the signature to **Amazon Cognito** for verification.
3. **Amazon Cognito** validates if the signature has been correctly signed by your wallet. If yes, a new session is established, and temporary AWS credentials are vended to the application.
4. Using the valid credentials, the dApp is now authorized to call the **Amazon API Gateway** endpoint to connect to a third-party API.
5. An **AWS Lambda** function can be used as backend integration for complex logic or if you need to make multiple underlying API calls to the third-party APIs.
6. Direct proxy calls can be made through **API Gateway**. No **Lambda** function is executed. API Keys* are injected impromptu using the Mapping Template.
7. Graphical assets are stored on InterPlanetary File System (IPFS) or HTTP.**

* The private API Keys of our Web3 providers are never exposed to the dApp and remain on the backend.

** IPFS is a storage protocol that is ideal for serving non-fungible tokens (NFTs) assets. HTTP is not ideal to serve NFTs assets, as it ties them to your host and domain name. You can also host the dApp itself on IPFS.

Reviewed for technical accuracy April 12, 2024

AWS Reference Architecture